

MC102 — Aula 02

Introdução à Programação de Computadores

Instituto de Computação – Unicamp

Primeiro Semestre de 2012

Roteiro

1 Organização de um computador

2 Algoritmos

3 A linguagem C

O que é um computador?

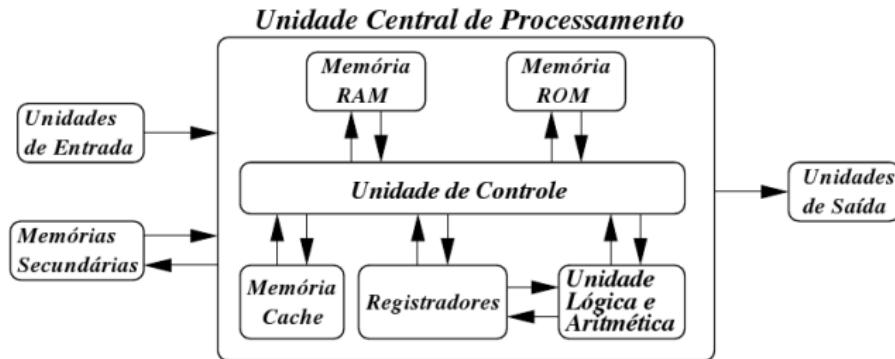
- Computador: o que computa, calculador, calculista. (dicionário Houaiss).
- Os primeiros “computadores” eram humanos que calculavam tabelas de logaritmos ou trajetórias pra canhões, seguindo procedimentos bem definidos.
- Um computador é uma máquina que, a partir de uma entrada, realiza um número muito grande de cálculos matemáticos e lógicos, gerando uma saída.

O que é um computador?

- O computador é um aparelho eletrônico, que por si só, não passa de monte de sucata.
- Para que possa ser útil é preciso programá-lo.
- Os programas de computadores ou *softwares* são os responsáveis pelo funcionamento e utilidade do equipamento.

Organização básica de um computador sequêncial

Um computador é uma coleção de componentes que realizam operações lógicas e aritméticas sobre um grande volume de dados.



Hardware e dispositivos

A linguagem nativa do computador é codificada numericamente, de forma binária:

- Bit → Pode assumir valores 0 ou 1.
- Byte → Agrupamento de 8 bits em uma palavra.
- Letras e símbolos são representados por números.

O que é um programa de computador?

- É um conjunto de instruções que orienta o computador a realizar uma determinada tarefa.
- Uma receita culinária orienta o cozinheiro na preparação de um prato.
- Um programa de computador (*softwares*) orienta o computador para a execução de uma tarefa.

Organização básica de um ambiente computacional

- Computadores realizam tarefas complexas por meio de um número enorme de operações simples.
- Para gerenciar a complexidade das soluções, existe uma hierarquia de funções, onde cada uma apresenta uma interface mais simples.

Programas de Aplicação		
Shell	Compiladores	Editores
Sistema operacional		
Hardware		

Programando computadores

- Como usuários, interagimos com os programas de aplicação.
- Neste curso iremos descer nesta hierarquia, para construirmos novos programas de aplicação.
- Estaremos interessados em **algoritmos** e em **linguagens de programação**, particularmente a linguagem **C**.

Lógica de Programação

- É a técnica utilizada para organizar e ordenar o pensamento com o objetivo de se obter uma solução para um determinado problema.
- Problema: trocar a lâmpada queimada do seu quarto.

Lógica de Programação

- A idéia é criar uma sequência de instruções claras e precisas, que representem a solução do problema sem dar margens à interpretações ambíguas.
- A esta sequência de instruções damos o nome de algoritmo.

Algoritmos

- Seqüência de passos, precisos e bem definidos, para a realização de uma tarefa.
- Algoritmos podem ser especificados de várias formas, inclusive em português.

Algoritmos

Exemplos de algoritmos

- Uma receita culinária;
- Um manual de instalação de um aparelho eletrônico;
- Um processo matemático para resolver um cálculo;
- Um roteiro para se chegar a um determinado local;
- Um programa de computador.

Algoritmos

- Todo algoritmo parte de uma situação inicial (**entrada**) e diz o que deve ser feito para se chegar a uma situação desejada (**saída**).
- Ingredientes - receita - bolo;
- Dados 4 valores - roteiro no processo de cálculo - responder com a média aritmética;

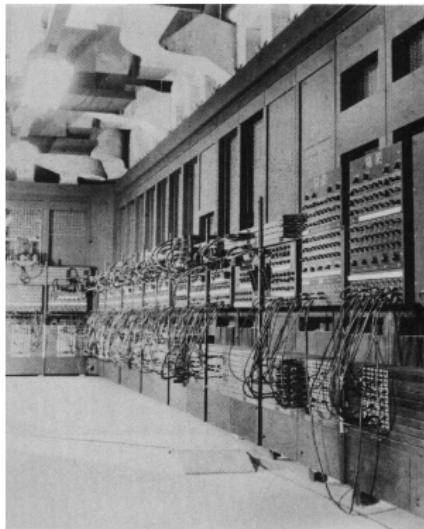
De algoritmos a programas

- Como transformar um algoritmo em linguagem que o computador entenda?
- Deve ser capaz de expressar tudo o que o computador pode fazer.
- Não pode ser ambígua.

Um pouco de história

Programação em código absoluto ou binário (apenas 0s e 1s).

ENIAC



Um pouco de história

Programação em código absoluto ou binário (apenas 0s e 1s).

```
0100 0010 0011 0101 0101 0100 0011 0110
0100 1110 1100 1100 1001 0110 0110 1000
0000 0101 1111 1110 1101 0011 0000 1100
```

Um pouco de história

Uma melhoria: A Linguagem Assembly.

- Cria-se uma linguagem de baixo nível (Linguagem Assembly) para representar as instruções em código binário.
- Um programa, chamado montador ou assembler, faz a transformação em código absoluto.

LOOP:	ADD	c,	a,	b	0100	0010	0011	0101	0101	0100	0011
ADD	i,	i,	1		0110	0110	0111	0101	0101	0100	0011
BNQ	i,	10,	loop		1111	0000	0111	0101	0101	0100	0011

Linguagens de alto nível

- Mais distantes da máquina e mais próximas de linguagens naturais (inglês, português, etc.).
- Mesmo mais comprehensíveis, elas não são ambíguas.
- Neste curso estamos interessados no estudo da *linguagem de programação C*.

Exemplos de linguagens

- C
- Pascal
- Java

```
for(i=0; i< 10; i++)  
    c = a+b;
```

Compiladores e Linguagens de Programação.

- Um *compilador* é um programa que lê um código de uma linguagem de programação e o transforma em um programa executável.
- Na verdade o compilador realiza esta tarefa juntamente com um *assembler*.

```
for(i=0; i< 10; i++)      loop:  add c, a, b      0100 0010 0011 0101 0101 0100  
      c = a+b;                add i, i,1      0110 0110 0111 0101 0101 0100  
                                bnq i, 10, loop 1111 0000 0111 0101 0101 0100
```

Primeiro programa em C

Um programa em C é um arquivo texto, contendo declarações e operações da linguagem. Isto é chamado de **código fonte**.

```
#include <stdio.h>

main() {
    printf("Hello, world!\n");
}
```

Como executar este programa

Para executar um programa a partir do seu código fonte é necessário compilá-lo, gerando **código binário** ou **executável**. Este pode ser executado como qualquer outro programa de aplicação.

```
$ gcc hello.c -o hello
$ hello
Hello, world!
```

O que são erros de compilação?

Caso o programa não esteja de acordo com as regras da linguagem, erros de compilação ocorrerão. **Ler** e entender estes erros é muito importante.

```
#include <stdio.h>
main() {
    printf("Hello, world!\n");
```

```
$ gcc hello.c -o hello
hello.c: In function ‘main’:
hello.c:5: error: syntax error at end of input
```

O que são erros de execução?

Acontecem quando o comportamento do programa diverge do esperado e podem acontecer mesmo quando o programa compila corretamente.

```
#include <stdio.h>
main() {
    printf("Hello, world! %#%#%@\n");
}
```

```
$ gcc hello.c -o hello
$ hello
Hello, world! %#%#%@%
```

Um exemplo mais complexo

```
#include <stdio.h>
main() {
    int x, y;
    printf("x: ");
    scanf("%d", &x);
    printf("y: ");
    scanf("%d", &y);
    if (x > y)
        printf ("O maior número é x = %d\n", x);
    else
        printf ("O maior número é y = %d\n", y);
}
```

Relembrando

- Hardware e Software.
- Pilha de um ambiente computacional: Programas Aplicações, Compilador, Sistema Operacional, Hardware.
- Código Binário, Assembly, Linguagem Alto Nível.
- Algoritmos.
- Erros de Compilação e Erros de execução.